



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/783,660	02/14/2001	Peter M. Mansour	SPRODQ1100	9105

29585 7590 06/17/2005

DLA PIPER RUDNICK GRAY CARY US LLP
153 TOWNSEND STREET
SUITE 800
SAN FRANCISCO, CA 94107-1907

EXAMINER

ZHONG, CHAD

ART UNIT	PAPER NUMBER
----------	--------------

2152

DATE MAILED: 06/17/2005

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.		Applicant(s)	
	09/783,660		MANSOUR ET AL.	
	Examiner		Art Unit	
	Chad Zhong		2154	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

- 1) ☒ Responsive to communication(s) filed on 29 March 2005.
- 2a) ☐ This action is **FINAL**. 2b) ☒ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

- 4) ☒ Claim(s) 1-70 is/are pending in the application.
- 4a) Of the above claim(s) _____ is/are withdrawn from consideration.
- 5) ☐ Claim(s) _____ is/are allowed.
- 6) ☒ Claim(s) 1-70 is/are rejected.
- 7) ☐ Claim(s) _____ is/are objected to.
- 8) ☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☐ The drawing(s) filed on _____ is/are: a) ☐ accepted or b) ☐ objected to by the Examiner.
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All b) ☐ Some * c) ☐ None of:
1. ☐ Certified copies of the priority documents have been received.
2. ☐ Certified copies of the priority documents have been received in Application No. _____.
3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

- | | |
|---|---|
| 1) <input checked="" type="checkbox"/> Notice of References Cited (PTO-892) | 4) <input type="checkbox"/> Interview Summary (PTO-413) |
| 2) <input type="checkbox"/> Notice of Draftsperson's Patent Drawing Review (PTO-948) | Paper No(s)/Mail Date. _____ |
| 3) <input type="checkbox"/> Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) | 5) <input type="checkbox"/> Notice of Informal Patent Application (PTO-152) |
| Paper No(s)/Mail Date _____ | 6) <input type="checkbox"/> Other: _____ |

DETAILED ACTION

1. This action is responsive to communications: RCE Amendment, filed on 03/29/2005.
2. Claims 1-70 are presented for examination. In RCE amendment, filed on 03/29/2005:
Claims 1, 38, 53, 66, 68, 70 are amended.

Applicant's remarks filed 03/29/2005 have been considered but are found not persuasive in view at the new grounds at rejection necessitated by Applicant's amendment.

Double Patenting

Claims 1-3, 5-6, 10-18, 46-47, 19-20, 29-31, 33-34, 36, 38, 45 of this application conflict with claims 1-53 of Application No. 09-783673. 37 CFR 1.78(b) provides that when two or more applications filed by the same applicant contain conflicting claims, elimination of such claims from all but one application may be required in the absence of good and sufficient reason for their retention during pendency in more than one application. Applicant is required to either cancel the conflicting claims from all but one application or maintain a clear line of demarcation between the applications. See MPEP § 822.

The nonstatutory double patenting rejection is based on a judicially created doctrine grounded in public policy (a policy reflected in the statute) so as to prevent the unjustified or improper timewise extension of the "right to exclude" granted by a patent and to prevent possible harassment by multiple assignees. See *In re Goodman*, 11 F.3d 1046, 29 USPQ2d 2010 (Fed. Cir. 1993); *In re Longi*, 759 F.2d 887, 225 USPQ 645 (Fed. Cir. 1985); *In re Van Ornum*, 686 F.2d 937, 214 USPQ 761 (CCPA 1982); *In re Vogel*, 422 F.2d 438, 164 USPQ 619 (CCPA 1970); and, *In re Thorington*, 418 F.2d 528, 163 USPQ 644 (CCPA 1969).

A timely filed terminal disclaimer in compliance with 37 CFR 1.321(c) may be used to overcome an actual or provisional rejection based on a nonstatutory double patenting ground provided the conflicting application or patent is shown to be commonly owned with this application. See 37 CFR 1.130(b).

Effective January 1, 1994, a registered attorney or agent of record may sign a terminal disclaimer. A terminal disclaimer signed by the assignee must fully comply with 37 CFR 3.73(b).

Claims 1-3, 5-6, 10-18, 46-47, 19-20, 29-31, 33-34, 36, 38, 45 are rejected under the judicially created doctrine of obviousness-type double patenting as being unpatentable over claims 1-53 of U.S. Patent No. 09-783673 in view of Simonoff et al. US 6,078,322.

Present Application: 09-783660	Co-Pending Application: 09-783673
<p>1. A data processing method comprising: generating, with a client device, a particular form of a client-resident intermediate user interface (UI) for a server-based and client-side controlled application according to a UI format determined by a UI server, including supplementing a skeletal UI stored in a first memory location with one or more icons, labels or menu items, or combinations thereof, stored in a second memory location. wherein the first memory location and the second memory location are situated on said client device; transmitting a number of source data items related to said server-based application from said UI server to said client device; and populating at least one native UI control used by said intermediate UI with said number of source data items.</p>	<p>1. A data processing method comprising: generating, with a client device, a particular form of a client-resident intermediate user interface (UI) for a server-based and client-side controlled application according to a UI format that is based upon a number of device capabilities for said client device, including supplementing a skeletal UI stored in a first memory location with one or more icons, labels or menu items, or combinations thereof, stored in a second memory location, wherein the first memory location and the second memory location are situated on said client device; receiving, at said client device, a number of source data items related to said server-based application; and populating at least one native UI control used by said intermediate UI with said member of source data items.</p>

Current application does not explicitly teach “transmitting a number of source data items related to said server-based application from said UI server to said client device”, Simonoff teaches the above see for example, (Col. 7, lines 23-30; Col. 9, lines 33-50; Col. 11, lines 60-67) in order to partially updating the UI in steps in accordance with the GUIScript objects.

Present Application: 09-783660	Co-Pending Application: 09-783673
3. a method according to claim 1, wherein said at least one native UI control is associated with an operating system for said client device.	2. A method according to claim 1, wherein said at least one native UI control is associated with an operating system for said client device.

Present Application: 09-783660	Co-Pending Application: 09-783673
5. a method according to claim 1, further comprising the steps of: generating an action request in response to a manipulation of said intermediate UI by a user of said client device; and updating said intermediate UI in response to said action request.	3. a method according to claim 1, further comprising the steps of: generating an action request in response to a manipulation of said intermediate UI by a user of said client device; and updating said intermediate UI in response to said action request.

Present Application: 09-783660	Co-Pending Application: 09-783673
6. a method according to claim 1, further comprising the steps of: performing an offline action by said client device while said client device is disconnected from said UI server; subsequently establishing a session between said client device and said UI server; and thereafter transmitting, from said client device to said UI server, a command	4. a method according to claim 1, further comprising the steps of: performing an offline action by said client device while said client device operates in a disconnected mode; subsequently establishing a session between said client device and a UI server; and thereafter transmitting, from said client device to said UI server, a command indicative of

indicative of said offline action.	said offline action.
---------------------------------------	----------------------

Present Application: 09-783660	Co-Pending Application: 09-783673
10. a method according to claim 1, further comprising this step of saving said number of source data items in a client cache resident at said client device.	5. a method according to claim 1, further comprising the step of saving said number of source data items in a client cache resident at said client device.

Present Application: 09-783660	Co-Pending Application: 09-783673
11. a method according to claim 10, further comprising the step of removing client cache items to accommodate said number of source data items.	6. a method according to claim 5, further comprising the step of removing client cache items to accommodate said number of source data items.

Present Application: 09-783660	Co-Pending Application: 09-783673
12. a method according to claim 11, wherein said removing step selectively removes said client cache items according to a hierarchical preference scheme.	7. a method according to claim 6, wherein said removing step selectively removes said client cache items according to a hierarchical preference scheme.

Present Application: 09-783660	Co-Pending Application: 09-783673
13. a method according to claim 1, further comprising the steps of: sending a client action command related to said server-based application from said UI server to said client device; and executing said client action command by said client device.	8. a method according to claim 1, further comprising the steps of: receiving, at said client device, a client action command related to said server-based application; and executing said client action command by said client device.

Present Application: 09-783660	Co-Pending Application: 09-783673
14. a method according to claim 1, wherein said number of source data items represent a portion of a larger amount of related data available at said UI server.	9. a method according to claim 1, wherein said number of source data items received during said receiving step represent a portion of a larger amount of related data available at a UI server.

Present Application: 09-783660	Co-Pending Application: 09-783673
15. a method according to claim 14, wherein: said larger amount of related data comprises a list of items; and said number of source data items represents a subset of said list of items.	10. (Original) A method according to claim 9 wherein: said larger amount of related data comprises a list of items; and said number of source data items represents a subset of said list of items.

Present Application: 09-783660	Co-Pending Application: 09-783673
16. a method according to claim 14, wherein: said larger amount of related data comprises a document; and said number of source data items represents a portion of said document.	11. a method according to claim 9, wherein: said larger amount of related data comprises a document; and said number of source data items represents a portion of said document.

Present Application: 09-783660	Co-Pending Application: 09-783673
17. a method according to claim 14, wherein: said larger amount of related data comprises an image; and said number of source data items represents a portion of said image.	12. a method according to claim 9, wherein: said larger amount of related data comprises an image; and said number of source data items represents a portion of said image.

Present Application: 09-783660	Co-Pending Application: 09-783673
18. a method according to claim 14, wherein: said larger amount of related data comprises a body of text; and said number of source data items represents a portion of said body of text.	13. a method according to claim 9, wherein: said larger amount of related data comprises a body of text; and said number of source data items represents a portion of said body of text.

Present Application: 09-783660	Co-Pending Application: 09-783673
46. a method according to claim 45, further comprising the step of specifying a command script corresponding to a manipulation of a UI control contained in said UI form, said command script being configured for execution by said client device.	14. a method according to claim 1, further comprising the step of retrieving a command script corresponding to a manipulation of a UI control contained in said intermediate UI, said command script being configured for execution by said client device

Present Application: 09-783660	Co-Pending Application: 09-783673
47. a method according to claim 46, further comprising the step of executing, by said client device, said command script in response to the manipulation of said UI control at said client device.	15. a method according to claim 14, further comprising the step of executing, by said client device, said command script in response to the manipulation of said UI control at said client device.

Present Application: 09-783660	Co-Pending Application: 09-783673
19. a data processing method comprising: defining a user interface (UI) form in response to a number of device capabilities for a client device; storing said UI form locally at said client device; saving a number of source data items locally at said client device, said number of source data items being related to a server-based	17. a data processing method comprising: storing a user interface (UI) form definition locally at a client device, said UI form definition being dictated by a number of device capabilities for said client device; said client device saving a number of source data items locally, said number of source data items being related to a served-based application;

<p>application executed by a UI server; and populating said UI form with said number of source data items, and wherein said number of source data items comprises a smaller subset than a total number of source data items related to said server-based application, and wherein further subsets of said total number of source data items are downloadable based upon execution of one or more client-side controls.</p>	<p>said client device rendering a UI that is based upon said UI form definition; and said client device populating said UI with said number of source data items, and wherein said number of source data items comprises a smaller subset than a total number of source data items related to said server-based application, and wherein further subsets of said total number of source data items are downloadable based upon execution of one or more client-side controls.</p>
--	--

Present Application: 09-783660	Co-Pending Application: 09-783673
20. a method according to claim 19, further comprising the step of transmitting said number of source data items from said UI server to said client device.	18. a method according to claim 17, further comprising the step of receiving, at said client device, said number of source data items from a UI server.

Present Application: 09-783660	Co-Pending Application: 09-783673
29. a method according to claim 28, wherein said removing step selectively removes said existing client cache items according to a hierarchical preference scheme.	23. a method according to claim 22, wherein said removing step selectively removes said client cache items according to a hierarchical preference scheme.

Present Application: 09-783660	Co-Pending Application: 09-783673
30. a method according to claim 27, further comprising the steps of: updating said UI form in response to a manipulation of a display control rendered by said client device; requesting an additional number of source data items from said UI server if said manipulation of said display control triggers a data request command; and	24. a method according to claim 21, further comprising the steps of: updating said UI in response to a manipulation of a display control rendered by said client device; requesting an additional number of source data items if said manipulation of said display control triggers a data request command; and replacing source data items saved in said client cache with said additional number of

replacing source data items saved in said client cache with said additional number of source data items.	source data items.
--	--------------------

Present Application: 09-783660	Co-Pending Application: 09-783673
31. a method according to claim 27, further comprising the steps of: updating said UI form in response to a manipulation of a display control rendered by said client device; retrieving additional source data items from said client cache in response to said manipulation of said display control; and displaying said additional source data items in said UI form.	25. a method according to claim 21, further comprising the steps of: updating said UI in response to a manipulation of a display control rendered by said client device; retrieving additional source data items from said client cache in response to said manipulation of said display control; and displaying said additional source data items in said UI.

Present Application: 09-783660	Co-Pending Application: 09-783673
33. a method according to claim 19, wherein said defining step defines said UI form based upon said server-based application.	27. a method according to claim 17, wherein said UI form definition is dictated by said server-based application.

Present Application: 09-783660	Co-Pending Application: 09-783673
34. a method according to claim 19, wherein said defining step defines said UI form with at least one native UI control stored locally at said client device.	28. a method according to claim 17, wherein said UI form definition identifies at least one native UI control stored locally at said client device.

Present Application: 09-783660	Co-Pending Application: 09-783673
36. a method according to claim 35, further comprising the steps of: said UI server receiving a request for additional source data items', and said UI server transmitting a subsequent portion of said total number of source data	30. a method according to claim 29, further comprising the steps of: said client device generating a request for additional source data items; and said client device receiving, from said UI server, a subsequent portion of said total

items to said client device in response to said request.	number of source data items.
--	------------------------------

Present Application: 09-783660	Co-Pending Application: 09-783673
<p>38. a data processing method comprising: executing, at a user interface (UI) server, a server-based application configured to manipulate source data items for presentment at a client device; displaying a particular UI form of a client-resident intermediate UI at said client device according to a UI format determined by a UI server, including supplementing a skeletal UI stored in a first memory location with one or more icons, labels or menu items, or combinations thereof stored in a second memory location, said UI form being capable of presenting data items to a user of said client device. wherein the first memory location and the second memory location are situated on said client device; generating a client-side controlled action request in response to a manipulation of said UI form by a user of said client device; and updating said UI form in response to said action request. associated with an operating system for said client device, including supplementing a skeletal UI stored in a first memory location with one or more icons, labels or menu items, or combinations thereof, stored in a second memory location wherein the first memory location and the second memory location are situated on said client device; said client device obtaining a number of data items related to said server-based application; and said client device displaying said number of data items in said at least one native UI control.</p>	<p>36. a client device architecture for use with a client device capable of communicating with a data processing server, said client device architecture comprising: a receive module configured to receive an instruction that identifies a user interface (UI) form definition; an operating system; a number of native UI controls provided by said operating system; a UI form data cache configured to store said UI form definition; and a UI module configured to generate a particular UI form of a client-resident intermediate UI for a server-based application according to said UI form definition, including supplementing a skeletal UI stored in a first memory location with one or more icons, labels or menu items, or combinations thereof, stored in a second memory location wherein the first memory location and the second memory location are situated on said client device. and to populate at least one of said native UI controls with a number of source data items associated with said server-based application.</p>

Present Application: 09-783660	Co-Pending Application: 09-783673
<p>45. a data processing method comprising: generating a user interface (UI) form definition for a server-based application based upon a number of device capabilities for a client device; instructing said client device to render a UI form corresponding to said UI form definition; rendering said UI form with at least one native UI control associated with an operating system for said client device; transmitting a number of data items from a UI server to said client device, said number of data items being related to said server-based application; and displaying said number of data items in said at least one native UI control, and wherein said number of source data items comprises a smaller subset than a total number of source data items related to said server-based application, and wherein further subsets of said total number of source data items are downloadable based upon execution of one or more client-side controls.</p>	<p>36. a client device architecture for use with a client device capable of communicating with a data processing server, said client device architecture comprising: a receive module configured to receive an instruction that identifies a user interface (UI) form definition; an operating system; a number of native UI controls provided by said operating system; a UI form data cache configured to store said UI form definition; and a UI module configured to generate a particular UI form of a client-resident intermediate UI for a server-based application according to said UI form definition, including supplementing a skeletal UI stored in a first memory location with one or more icons, labels or menu items, or combinations thereof, stored in a second memory location wherein the first memory location and the second memory location are situated on said client device. and to populate at least one of said native UI controls with a number of source data items associated with said server-based application.</p>

Present Application: 09-783660	Co-Pending Application: 09-783673
<p>2. a method according to claim 1, further comprising the step of formatting characteristics of said intermediate UI based upon a number of device capabilities for said client device.</p>	<p>44. a client device architecture according to claim 36, wherein said UI form definition is based upon a number of device capabilities for said client device.</p>

Claim Rejections - 35 USC § 102

3. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

A person shall be entitled to a patent unless --

(e) the invention was described in a patent granted on an application for patent by another filed in the United States before the invention thereof by the applicant for patent, or on an international application by another who has fulfilled the requirements of paragraphs (1), (2), and (4) of section 371 (c) of this title before the invention thereof by the applicant for patent.

5. Claims 1-5, 9-11, 12, 13-17, 19-23, 27-28, 29, 30-64, 65-70 are rejected under 35 U.S.C. 102(e) as being anticipated by Simonoff et al. (hereinafter Simonoff), US 6,327,608.

6. As per claim 1, Simonoff teaches a data processing method comprising:

generating, with a client device (universal client), a particular form of a client-resident intermediate user interface (UI) for a server-based and client-side controlled application according to a UI format determined by a UI server (server 100; Col. 9, lines 40-50), including supplementing a skeletal UI stored in a first memory location (the GUI script file, Col. 9, lines 35-50) with one or more icons, labels or menu items, or combinations thereof, stored in a second memory location (Col. 9, lines 33-50; Col. 11, lines 60-67), wherein the first memory location and the second memory location are situated on said client device (Col. 14, lines 35-67);

transmitting a number of source data items related to said server-based application from said UI server to said client device (Col. 9, lines 33-50); and

populating at least one native UI control used by said UI with said number of source data items (Col. 9, lines 33-50; Col. 14, lines 33-41; Col. 16, lines 40-49).

7. As per claim 2, Simonoff teaches a method according to claim 1, further comprising the step of formatting characteristics of said intermediate UI based upon a number of device capabilities for said client device (Col. 7, lines 35-41).

8. As per claim 3, Simonoff teaches a method according to claim 1, wherein said at least one native UI control is associated with an operating system for said client device (Col. 11, lines 64-67).

9. As per claim 4, Simonoff teaches a method according to claim 1, further comprising the step of executing, at said UI server, said server-based application to manipulate source data items for presentment at said client device (Col. 9, lines 33-50).

10. As per claim 5, Simonoff teaches a method according to claim 1, further comprising the steps of:

generating an action request in response to a manipulation of said UI by a user of said intermediate client device (Col. 12, lines 1-13); and

updating said intermediate UI in response to said action request (Col. 12, lines 1-13).

11. As per claim 9, Simonoff teaches a method according to claim 1, further comprising the step of maintaining a shadow cache at said UI server, said shadow cache including a list of source data items transmitted from said UI server to said client device (Col. 8, lines 11-19; Col. 9, lines 33-50).

12. As per claim 10, Simonoff teaches a method according to claim 1, further comprising the step of saving said number of source data items in a client cache resident at said client device (Col. 14, lines 44-56).

13. As per claim 11, Simonoff teaches a method according to claim 5, further comprising the step of removing client cache items to accommodate said number of source data items (Col. 13, lines 35-44; Col. 14, lines 44-56).

14. As per claim 13, Simonoff teaches a method according to claim 1, further comprising the

steps of:

sending a client action command related to said server-based application from said UI server to said client device; and

executing said client action command by said client device (Col. 9, lines 33-50).

15. As per claim 14, Simonoff teaches a method according to claim 1, wherein said number of source data items received during said receiving step represent a portion of a larger amount of related data available at a UI server (Col. 16, lines 40-49; Col. 14, lines 44-56; Col. 9, lines 33-50; Col. 10, lines 23-30).

16. As per claim 15, Simonoff teaches a method according to claim 14, wherein:

said larger amount of related data comprises a list of items; and said number of source data items represents a subset of said list of items (Col. 16, lines 40-49).

17. As per claim 16, Simonoff teaches a method according to claim 14, wherein:

said larger amount of related data comprises a document (Col. 9, lines 33-50); and

said number of source data items represents a portion of said document (Col. 9, lines 33-50; Col. 16, lines 40-49).

18. As per claim 17, Simonoff teaches a method according to claim 14, wherein:

said larger amount of related data comprises an image; and

said number of source data items represents a portion of said image (Col. 16, lines 40-49).

19. As per claim 19, Simonoff teaches a data processing method comprising:

defining a user interface (UI) form in response to a number of device capabilities for a client device (Col. 9, lines 33-50);

storing said UI form locally at said client device (Col. 10, lines 23-30);

saving a number of source data items locally at said client device (Col. 10, lines 23-30), said number of source data items being related to a server-based application executed by a UI server (Col. 9, lines 33-50; Col. 12, lines 1-13); and

populating said UI form with said number of source data items, and wherein said number of source data items comprises a smaller subset than a total number of source data items related to said server-based application, and wherein further subsets of said total number of source data items are downloadable based upon execution of one or more client-side controls (Col. 9, lines 33-50).

20. As per claim 20, Simonoff teaches a method according to claim 19, further comprising the step of transmitting said number of source data items from said UI server to said client device (Col. 9, lines 33-50).

21. As per claim 21, Simonoff teaches a method according to claim 19, wherein said defining step is performed by said UI server in response to a device identifier obtained from said client device (Col. 9, lines 8-25).

22. As per claim 22, Simonoff teaches a method according to claim 19, further comprising the step of executing, at said UI server, said server-based application to manipulate source data items for presentment at said client device (Col. 12, lines 1-13).

23. As per claim 23, Claim 23 is rejected for the same reasons as rejection to claim 5 above.

23. As per claims 27-28, claims 27-28 are rejected for the same reasons as rejection to claims 10-11 above respectively.

24. As per claim 30, Simonoff teaches a method according to claim 27, further comprising the steps

of:

updating said UI form in response to a manipulation of a display control rendered by said client device;

requesting an additional number of source data items from said UI server if said manipulation of said display control triggers a data request command; and

replacing source data items saved in said client cache with said additional number of source data items (Col. 12, lines 1-13).

25. As per claim 31, Simonoff teaches a method according to claim 27, further comprising the steps of:

updating said UI form in response to a manipulation of a display control rendered by said client device;

retrieving additional source data items from said client cache in response to said manipulation of said display control; and

displaying said additional source data items in said UI form (Col. 12, lines 1-13, lines 23-39).

26. As per claim 32, Claim 32 is rejected for the same reasons as rejection to claim 13 above.

27. As per claim 33, Simonoff teaches a method according to claim 19, wherein said defining step defines said UI form based upon said server-based application (Col. 9, lines 33-50).

28. As per claim 34, Simonoff teaches a method according to claim 19, wherein said defining step defines said UI form with at least one native UI control stored locally at said client device (Col. 10, lines 23-30; Col. 12, lines 23-39).

29. As per claim 35, Simonoff teaches a method according to claim 19, wherein:

Art Unit: 2152

said UI server has access to a total number of source data items associated with said UI form (Col. 9, lines 33-50); and

said number of source data items saved during said saving step represents a portion of said total number of source data items (this section of the claim 35 is rejected for the same reasons as rejection to claim 14 above).

30. As per claim 36, Simonoff teaches a method according to claim 35, further comprising the steps of:

said UI server receiving a request for additional source data items (Col. 12, lines 1-13; Col. 9, lines 33-50); and

said UI server transmitting a subsequent portion of said total number of source data items to said client device in response to said request (Col. 12, lines 1-13; Col. 9, lines 33-50).

31. As per claim 37, Simonoff teaches a method according to claim 36, wherein said UI server receives said request from said client device in response to a manipulation of said UI form (Col. 12, lines 1-13).

32. As per claim 38, Simonoff teaches a data processing method comprising:

executing, at a user interface (UI) server, a server-based application configured to manipulate source data items for presentment at a client device;

displaying a particular UI form of a client-resident intermediate UI at said client device according to a UI format determined by a UI server, including supplementing a skeletal UI stored in a first memory location with one or more icons, labels or menu items, or combinations thereof, stored in a second memory location, said UI form being capable of presenting data items to a user of said client device, wherein the first memory location and the second memory location are situated on said client device (Col. 14, lines 35-67);

generating a client side controlled action request in response to a manipulation of said UI form by a user of said client device; and

updating said UI form in response to said action request (Col. 9, lines 33-50).

33. As per claim 39, Simonoff teaches a method according to claim 38, further comprising the steps of:

sending said action request from said client device to said UI server; and

processing said action request by said UI server (Col. 9, lines 33-50; Col. 12, lines 1-13).

34. As per claim 40, Simonoff teaches a method according to claim 38, further comprising the step of transmitting a number of source data items related to said server-based application from said UI server to said client device, said transmitting step being performed in response to said action request (Col. 9, lines 33-50).

35. As per claim 41, Claim 41 is rejected for the same reasons as rejection to claim 14 above.

36. As per claim 42, Simonoff teaches a method according to claim 41, further comprising the steps of:

requesting, from said UI server, said number of source data items in response to an initial manipulation of said UI form; and

subsequently requesting, from said UI server, an additional number of source data items in response to a further manipulation of said UI form;

wherein said additional number of source data items represent a second portion of said larger amount of related data (Col. 9, lines 33-50; Col. 12, lines 1-40).

37. As per claim 43, Simonoff teaches a method according to claim 38, further comprising the steps

of:

said UI server receiving information representing new, deleted, or modified data items; and
said UI server transmitting, to said client device, push data representing said new, deleted, or modified source data items (Col. 9, lines 33-50).

38. As per claim 44, Simonoff teaches A method according to claim 43, further comprising the step of said UI server sending, to said client device, a push notification corresponding to said push data (Col. 12, lines 1-45).

39. As per claim 45, Simonoff teaches a data processing method comprising:

generating a user interface (UI) form definition for a server-based application based upon a number of device capabilities for a client device;

instructing said client device to render a UI form corresponding to said UI form definition;

rendering said UI form with at least one native UI control associated with an operating system for said client device;

transmitting a number of data items from a UI server to said client device, said number of data items being related to said server-based application; and

displaying said number of data items in said at least one native UI control, and wherein said number of source data items comprises a smaller subset than a total number of source data items related to said server-based application, and wherein further subsets of said total number of source data items are downloadable based upon execution of one or more client-side controls (Col. 9, lines 33-50; Col. 12, lines 1-40).

40. As per claim 46, Simonoff teaches a method according to claim 45, further comprising the step of specifying a command script corresponding to a manipulation of a UI control contained in said UI form, said command script being configured for execution by said client device (Col. 9, lines 33-50).

41. As per claim 47, Simonoff teaches a method according to claim 46, further comprising the step of executing, by said client device, said command script in response to the manipulation of said UI control at said client device (Col. 9, lines 33-50; Col. 12, lines 1-13).

42. As per claim 48, Simonoff teaches a method according to claim 45, further comprising the step of saving said number of data items in a client cache resident at said client device (Col. 14, lines 44-56).

43. As per claim 49, Simonoff teaches a method according to claim 48, further comprising the step of retrieving said number of data items from said client cache prior to said displaying step (Col. 14, lines 44-56; Col. 12, lines 1-40).

44. As per claim 50, Simonoff teaches a method according to claim 45, further comprising the step of requesting, from said UI server, said number of data items in response to a manipulation of said at least one native UI control (Col. 12, lines 1-13).

45. As per claim 51, Claim 51 is rejected for the same reasons as rejection to claim 14 above.

46. As per claim 52, Claim 52 is rejected for the same reasons as rejection to claim 42 above.

47. As per claim 53, Simonoff teaches a distributed user interface (UI) architecture comprising:
a client device architecture comprising a UI module configured to generate a particular form of a client-resident intermediate UI for a server-based and client-side controlled application according to a UI form definition, by supplementing a skeletal UI stored in a first memory location with one or more icons, labels or menu items, or combinations thereof, stored in a second memory location, and to populate at least one native UI control used by said intermediate UI with source data items, wherein the first memory location and the second memory location are situated in said client device architecture (Col. 14, lines 35-

67); and

a UI server architecture comprising a server send module configured to transmit, to said client device architecture, a number of source data items related to said server-based application;

wherein said UI module (Col. 15, lines 17-27) populates said UI control with said number of source data items (Col. 9, lines 33-50; Col. 12, lines 1-13).

48. As per claim 54, Simonoff teaches a distributed UI architecture according to claim 53, wherein said UI server architecture further comprises a UI formatting module that generates said UI form definition based upon a number of device capabilities for a client device that includes said client device architecture (Col. 9, lines 33-50).

49. As per claim 55, Simonoff teaches a distributed UI architecture according to claim 53, wherein said client device architecture further comprises a client cache configured to store said number of source data items (Col. 14, lines 44-56).

50. As per claim 56, Simonoff teaches A distributed UI architecture according to claim 55, wherein said UI server architecture further comprises a shadow cache configured to store data representing the contents of said client cache (Col. 8, lines 15-20).

51. As per claim 57, Simonoff teaches a distributed UI architecture according to claim 55, wherein said client cache is further configured to store said UI form definition (Col. 12, lines 44-56).

52. As per claim 58, Claim 58 is rejected for the same reasons as rejection to claim 14 above.

53. As per claim 59, Simonoff teaches a distributed user interface (UI) system comprising:

a client device having a client processing architecture and

a client communication element configured to communicate with a compatible communication

element; and

a UI server having a server processing architecture and a server communication element configured to communicate with said client communication element;

said client processing architecture being configured to:

transmit a device identifier to said UI server;

generate a UI form in accordance with a UI form definition; and

populate at least one native UI control with a number of source data items associated with a server-based application;

said server processing architecture being configured to:

receive said device identifier from said client device;

identify said UI form definition in response to said device identifier; and

send said number of source data items to said client device for rendering with said UI form, and

wherein said number of source data items comprises a smaller subset than a total number of source data items related to said server-based application, and wherein further subsets of said total number of source data items are downloadable based upon execution of one or more client-side controls (Col. 9, lines 33-50, lines 8-25; Col. 12, lines 1-40;).

54. As per claim 60, Simonoff teaches a system according to claim 59, wherein:

said client device includes a number of device capabilities related to UI characteristics; and

said server processing architecture is further configured to generate said UI form definition based upon said number of device capabilities (Col. 9, lines 33-50; Col. 11, lines 60-67).

55. As per claim 61, Simonoff teaches a system according to claim 59, wherein said client device further comprises a client cache configured to store said number of source data items (Col. 14, lines 46-56).

56. As per claim 62, Simonoff teaches a system according to claim 59, wherein said client device further comprises a client cache configured to store said UI form definition (Col. 14, lines 46-56; Col. 12, lines 1-40; Col. 10, lines 23-30).

57. As per claim 63, Claim 63 is rejected for the same reasons as rejection to claim 14 above.

58. As per claim 64, Claim 64 is rejected for the same reasons as rejection to claim 42 above.

59. As per claims 65-66, claims 65-66 are rejected for the same reasons as rejection to claim 19 and 1 above respectively.

60. As per claims 67-68, claims 65-66 are rejected for the same reasons as rejection to claim 19 and 1 above respectively.

61. As per claims 69-70, claims 65-66 are rejected for the same reasons as rejection to claim 19 and 1 above respectively.

62. As per claim 12, Simonoff teaches a method according to claim 11, wherein said removing step selectively removes said client cache items according to a hierarchical preference scheme (Col. 13, lines 35-52).

63. As per claim 29, Claim 29 is rejected for the same reasons as rejection to claim 12 above.

Claim Rejections - 35 USC § 103

64. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having

ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

65. Claim 18 is rejected under 35 U.S.C. 103(a) as being unpatentable over Simonoff et al. (hereinafter Simonoff), US 6,078,322, in view of 'Official Notice'.

66. As per claim 18, Simonoff does not teach a method according to claim 14, wherein:
said larger amount of related data comprises a body of text; and
said number of source data items represents a portion of said body of text. However 'Official Notice' is taken by the Examiner that a text file is notoriously well known as a type of file. It would have been obvious to have used a text file for the purpose of the current invention, because doing so would be less burdening for the individual units, through the usage of text file in place of image or a document and the like, the user now have the option of manipulating a portion of the text file thereby improving processing efficiency and speed on the client side, thus realizing a thin client network.

67. Claims 6-8, 24-26 are rejected under 35 U.S.C. 103(a) as being unpatentable over Simonoff et al. (hereinafter Simonoff), US 6,078,322, in view of "Browser wars: Rest in peace", Patrick, January 2001.

68. As per claim 6, Simonoff teaches a method according to claim 1, further comprising the steps of:

performing an offline action by said client device while said client device operates in a disconnected mode (Col. 10, lines 23-30);

subsequently establishing a session between said client device and a UI server (Col. 17, lines 10-15; wherein the client establishes with server if there is a connection failure or switch over to a back up server);

Art Unit: 2152

Simonoff does not explicitly teaches:

thereafter transmitting, from said client device to said UI server, a command indicative of said offline action.

Patrick teaches of a system wherein the users has the option of manipulating data while offline and the changed data gets updated when the software goes online again, in order to allow for an offline usage option for the users.

Patrick teaches:

thereafter transmitting, from said client device to said UI server, a command indicative of said offline action (pg 2, 2nd paragraph).

It would have been obvious to combine teachings of Simonoff and Patrick in order to provide the user to an offline manipulate and access of data (pg 1, 10th paragraph).

69. As per claim 7, Simonoff does not explicitly teach a method according to claim 6, further comprising the step of executing said command by said server-based application

Patrick teaches of a system wherein the users has the option of manipulating data while offline and the changed data gets updated when the software goes online again, in order to allow for an offline usage option for the users.

Patrick teaches:

a method according to claim 6, further comprising the step of executing said command by said server-based application (pg 2, 2nd paragraph, wherein peer-to-peer networks the clients are acting as servers exchange information between each other).

It would have been obvious to combine teachings of Simonoff and Patrick in order to provide the user to an offline manipulate and access of data (pg 1, 10th paragraph).

70. As per claim 8, Simonoff teaches a method according to claim 6, wherein:

said offline action modifies at least one of said source data items at said client device (Col. 10, lines 23-30); and

Simonoff does not explicitly teach:

said method further comprises the step of updating a corresponding number of source data items maintained by said UI server to reflect the modification of said source data items.

Patrick teaches of a system wherein the users has the option of manipulating data while offline and the changed data gets updated when the software goes online again, in order to allow for an offline usage option for the users.

Patrick teaches:

said method further comprises the step of updating a corresponding number of source data items maintained by said UI server to reflect the modification of said source data items (pg 1, 10th paragraph; pg 2, 2nd paragraph).

It would have been obvious to combine teachings of Simonoff and Patrick in order to provide the user to an offline manipulate and access of data (pg 1, 10th paragraph).

71. As per claims 24-26, Claims 24-26 are rejected for the same reasons as rejection to claims 6-8 above respectively.

Conclusion

72. In the remark, the Applicant argued in substance that Simonoff fails to disclose or suggest “indication in the reference that the GUIScript file or the GUI is anything less than an entire UI displayed to a user... GUI objects as being skeletal nor supplementing the same”

In response to Applicant’s arguments, Simonoff teaches the above limitations. See for example, Col. 12, lines 24-60, wherein the GUIScripts carries a GUI change in response to an event, the event, i.e. button clicking, has resulted in partial GUI updating, only a portion of the GUI are generated at a time, any

additional changes applies to the existing GUI are based upon system events, thus, GUI elements are supplemented in accordance with client requests, (see also, Col. 11, lines 55-67; Col. 12, lines 40-45).

Hence, Simonoff teaches the foregoing section.

73. In the remark, the Applicant argued in substance that Simonoff realizes the use of two memory locations only by pushing an entire GUI object "from the server to the client", however, Simonoff fails to disclose or suggest "both memory locations to be situated on the client device".

In response to Applicant's arguments, the Examiner agrees with the applicant that pushing GUI objects from server to client is taught by Simonoff. However, in another embodiment, Simonoff discloses of a system running as a stand alone architecture and carry out similar methodologies as the client server model, exchanging GUI objects, see for example, Col. 14, lines 33-67.

Hence, Simonoff teaches the newly amended section as presented above.

74. In the remark, the Applicant argued in substance that Simonoff fails to disclose or suggest "GUIScript populating a native UI control".

In response to Applicant's arguments, Simonoff teaches this limitation. See for example Fig 7, in a non-limiting example, Simonoff discloses of Chatroom GUI, in a chatroom, the GUI is updated when the user selects the event button, i.e. send/clear/close, send will display the new messages to users of the chatroom, thus only the portion of the GUI gets updated periodically in accordance to user actions, while remainder of the GUI remains fixed, native UI control is thus populated with user entries. Furthermore, in Col. 12, lines 25-67, the GUI gets an update upon a system event. The 'send' event would cause partial update of the GUI as shown in Fig 7, thus realizing GUIScript populating a native UI control.

Art Unit: 2152

75. In the remark, Applicant argued in substance that portions of claim 6-8, 24-26 are missing, please refer to the action above for additional details regarding those claims.

76. In the remark, Applicant argued in substance that Simonoff fails to disclose or suggest "UI format is determined by a UI server".

In response to Applicant's arguments, Simonoff teaches the above limitation. See for example, Col. 9, lines 35-50, wherein the server is able to select the format prior to sending to the client. Hence, Simonoff teaches the claimed section above.

77. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure. The following patents and publications are cited to further show the state of the art with respect to "PLATFORM-INDEPENDENT DISTRIBUTED USER INTERFACE CLIENT ARCHITECTURE".

- i. US 5818447 Wolf et al.
- ii. US 2002/0152244 Dean et al.
- iii. US 6167534 Straathof et al.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Chad Zhong whose telephone number is (571)272-3946. The examiner can normally be reached on M-F 7:15 to 4:30.

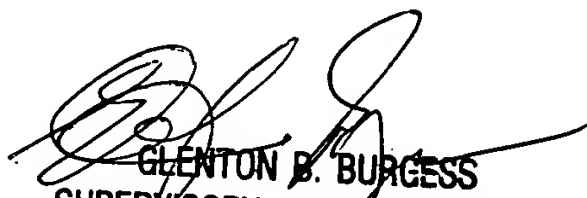
If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, BURGESS, GLENTON B can be reached on (571)272-3949. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available

Art Unit: 2152

through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

CZ
May 2, 2005



GLENTON B. BURGESS
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100